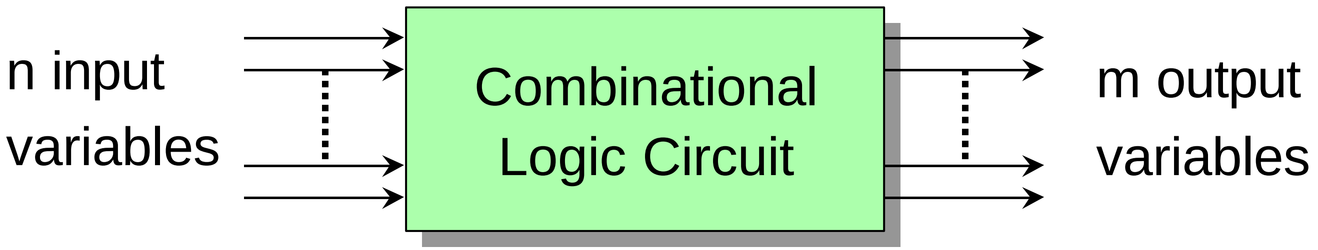
4장 Combinational Logic Circuit

Combinational Logic Circuit

❑ Digital Logic Circuit

- ◆ Combinational logic circuit)
  - 출력 = f(현재입력)
  - **현재의 입력에 의해서만 출력이 결정**
    n개의 입력이 일정시간후 m개의 출력을 생성
  - No feedback



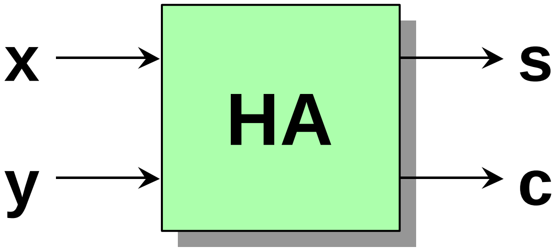
- ◆ Sequential logic circuit
  - 출력 = f(현재입력, 현재상태)
  - **현재의 입력뿐만 아니라 현재의 상태도 출력에 영향**
  - feedback

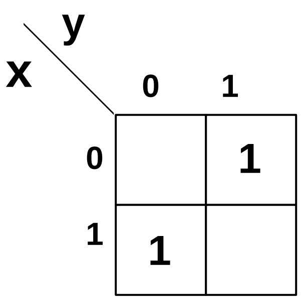
Design of Combinational Circuit

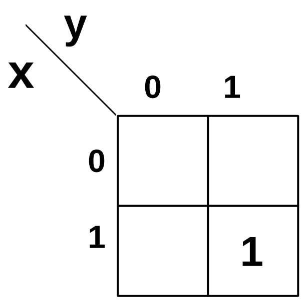
- Design Procedure
  - (1) 문제 기술
  - (2) Input, output variable의 갯수 결정
  - (3) Input, output variable의 이름 할당
  - (4) Truth table 작성 - 입력의 조합수를 파악하여 무관조건항 설정
  - (5) Boolean function의 minimization
  - (6) Logic diagram 작성
  - (7) Design verification

- 설계시 제약조건
  - 최소갯수의 논리게이트
  - 게이트당 최소입력수
  - 최소의 전달지연시간
  - 최소의 연결선 수
  - 각 논리게이트의 구동한계(fanout)

Half-Adder



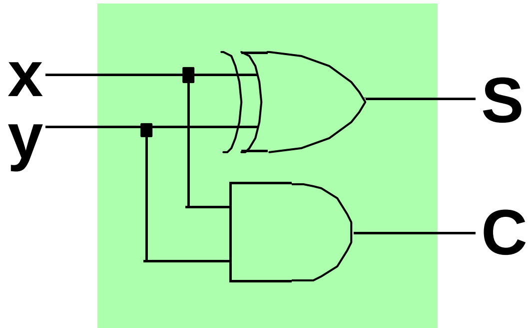




x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

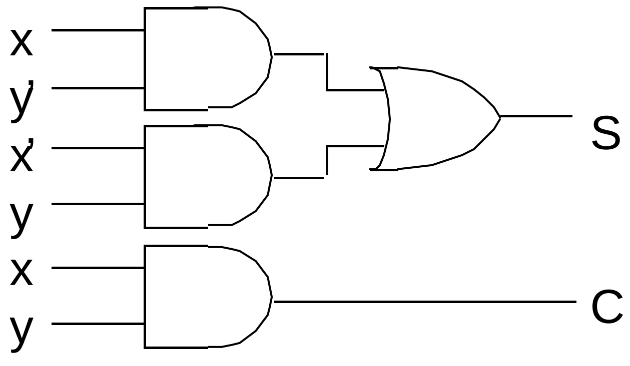
$$S = xy' + x'y$$
$$= x \oplus y$$

$$C = xy$$



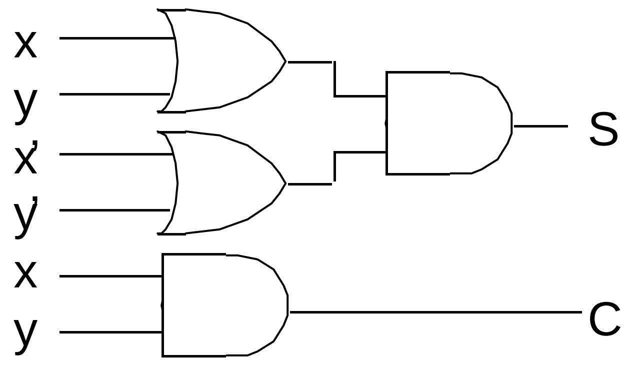
jhlee@hnu.ac.kr

Half-Adder의 구현



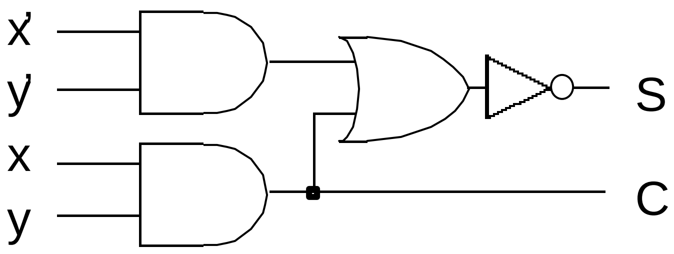
$S = xy' + x'y$

$C = xy$



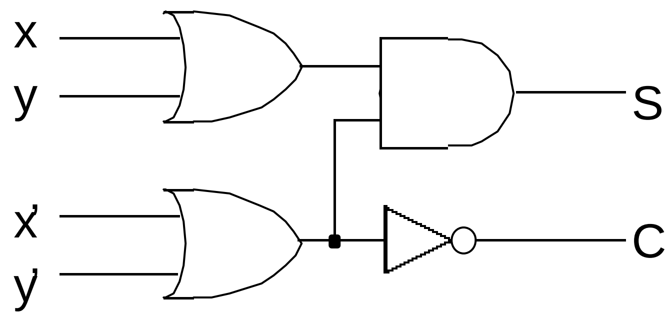
$S = (x+y)(x'+y')$

$C = xy$



$S = (C+x'y')'$

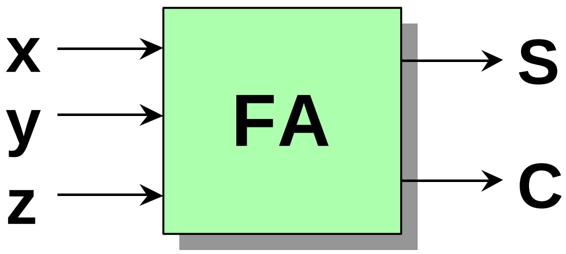
$C = xy$



$S = (x+y)(x'+y')$

$C = (x'+y')'$

Full-Adder



x	y	z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

x \ yz	00	01	11	10
0		1		1
1	1		1	

$$S = xy'z' + x'y'z + x'yz' + xyz$$
$$= x'(y'z + yz') + x(yz + y'z')$$
$$= x'(y \oplus z) + x(y \oplus z)'$$
$$= x \oplus y \oplus z$$

x \ yz	00	01	11	10
0			1	
1		1	1	1

$$C = xy + yz + xz$$

x \ yz	00	01	11	10
0			1	
1		1	1	1

$$C = xy + x'yz + xy'z$$
$$= xy + z(x'y + xy')$$
$$= xy + z(x \oplus y)$$

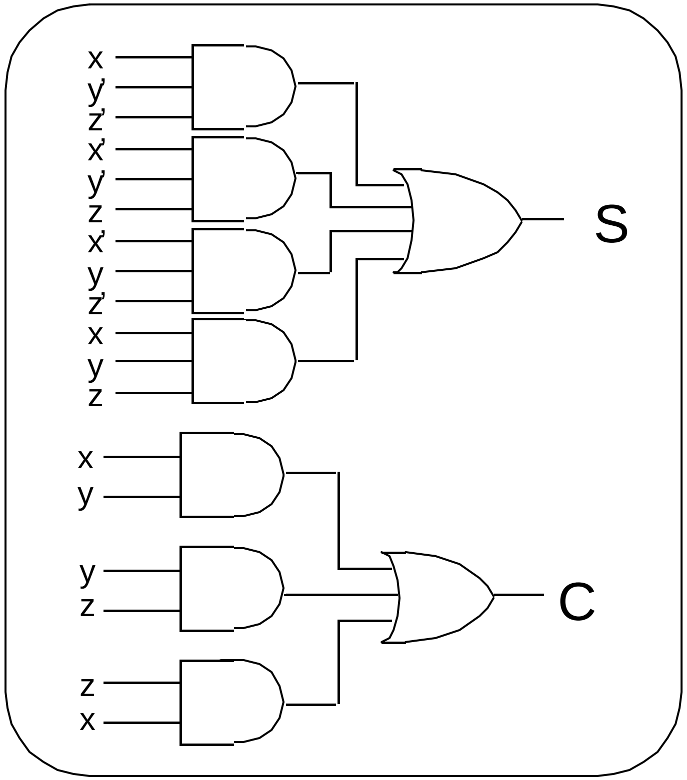
Full-Adder의 논리회로도

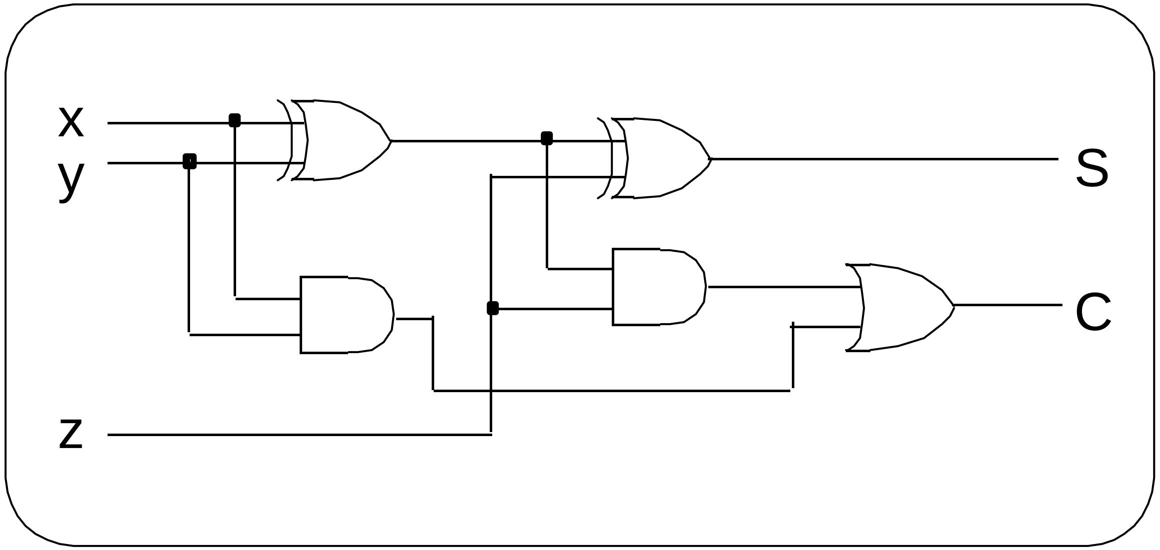
$S = xy'z' + x'y'z + x'yz' + xyz$

$C = xy + yz + xz$

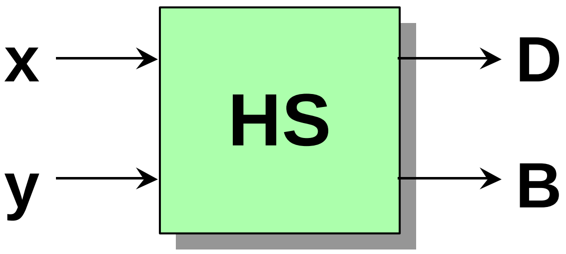
$S = x \oplus y \oplus z$

$C = xy + z(x \oplus y)$

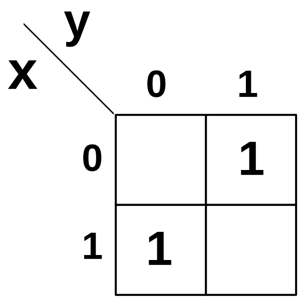


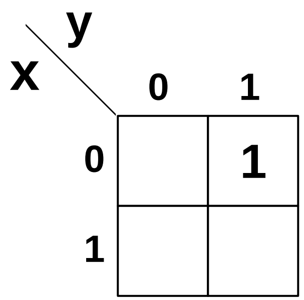


Half-Subtractor



x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

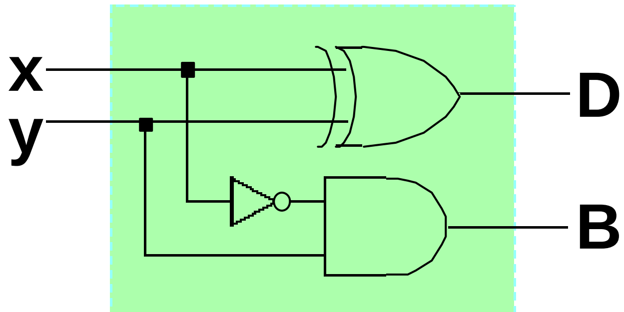




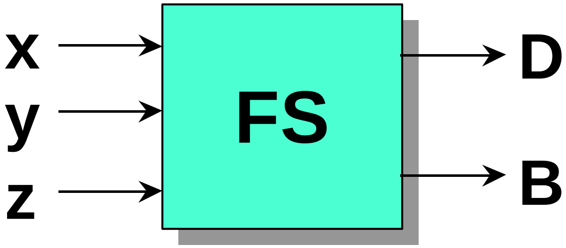
$D = xy' + x'y$

$= x \oplus y$

$B = x'y$



Full-Subtracter



x	y	z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

x \ yz	00	01	11	10
0		1		1
1	1		1	

$$D = xy'z' + x'y'z + x'yz' + xyz$$
$$= x'(y'z + yz') + x(yz + y'z')$$
$$= x'(y \oplus z) + x(y \oplus z)'$$
$$= x \oplus y \oplus z$$

x \ yz	00	01	11	10
0		1	1	1
1			1	

$$B = x'y + yz + x'z$$

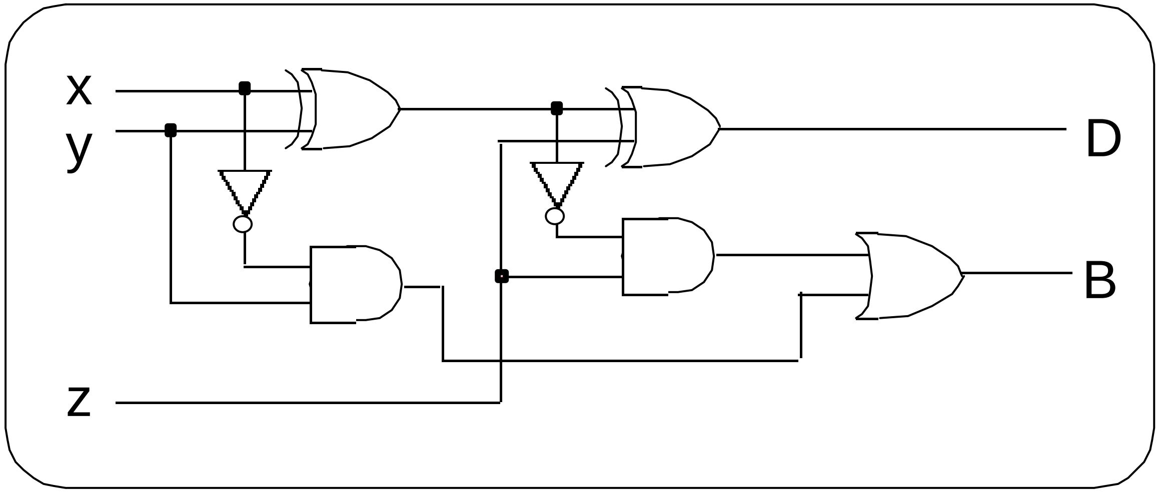
yz	00	01	11	10
0		1	1	1
1			1	

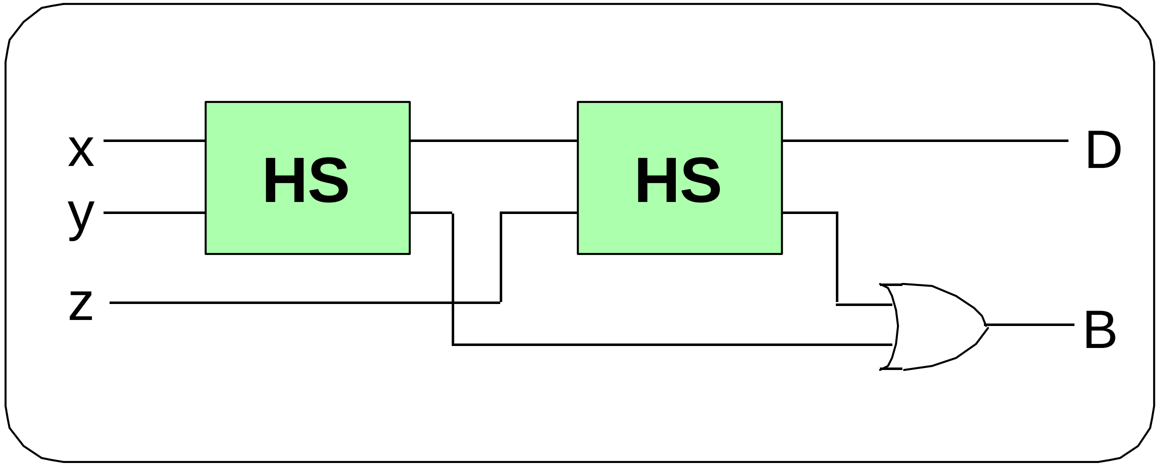
$$B = x'y + xyz + x'y'z$$
$$= x'y + z(xy + x'y')$$
$$= x'y + z(x \oplus y)'$$

Full-subtracter의 논리회로도

D = x⊕y⊕z

B = x' y+z(x⊕y)'





jhlee@hnu.ac.kr

BCD to Excess-3 Code Converter

A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

$w(A,B,C,D) = \sum(5,6,7,8,9)$

$x(A,B,C,D) = \sum(1,2,3,4,9)$

$y(A,B,C,D) = \sum(0,3,4,7,8,)$

$z(A,B,C,D) = \sum(0,2,4,6,8)$

$d(A,B,C,D) = \sum(10,11,12,13,14,15)$

AB \ CD	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$w = A+BC+BD$

AB \ CD	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

$x = B'C+B'D+BC'D'$

AB \ CD	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

$y = CD+C'D'$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$z = D'$

BCD to Excess-3 Code Converter의 논리회로도

❑ 출력함수의 변형

w = A+BC+BD

  = A+B(C+D)

x = B' C+B' D+BC' D'

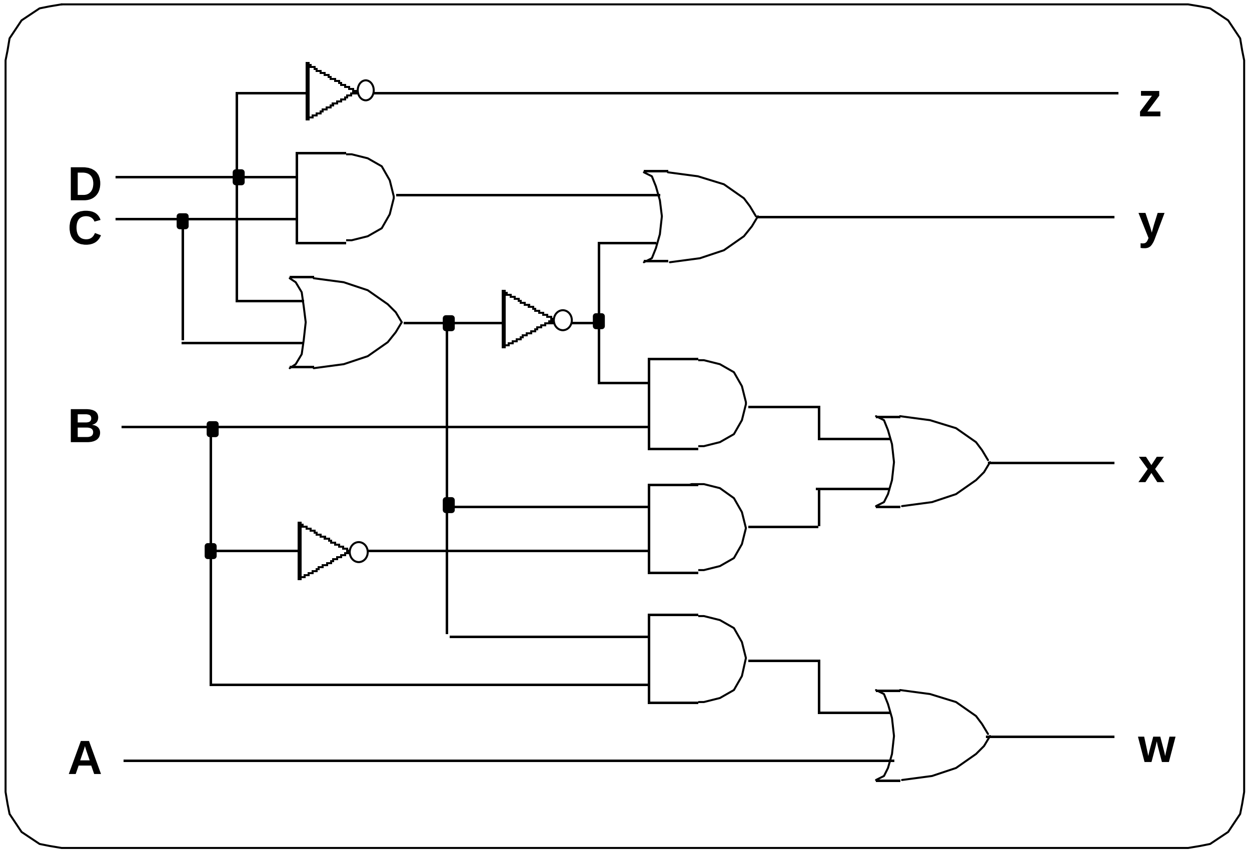
  = B' (C+D)+BC' D'

  = B' (C+D)+B(C+D)'

y = CD+C' D'

  = CD+(C+D)'

z = D'

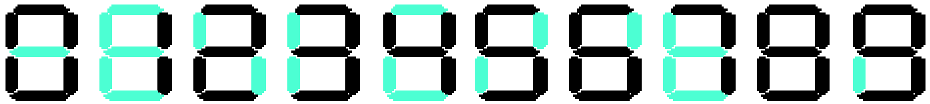


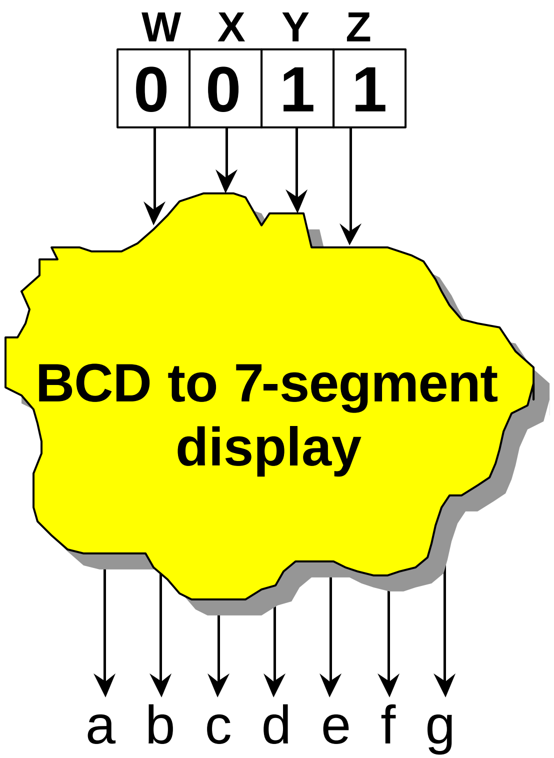
※ 식의 변형 전과 후의 게이트 수 비교

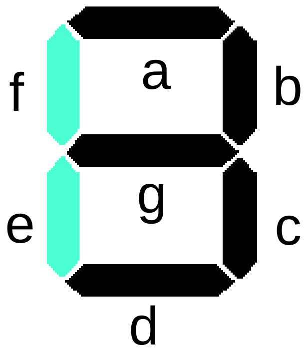
전 : AND(7), OR(3)

후 : AND(4), OR(4)

BCD to 7-Segment Display

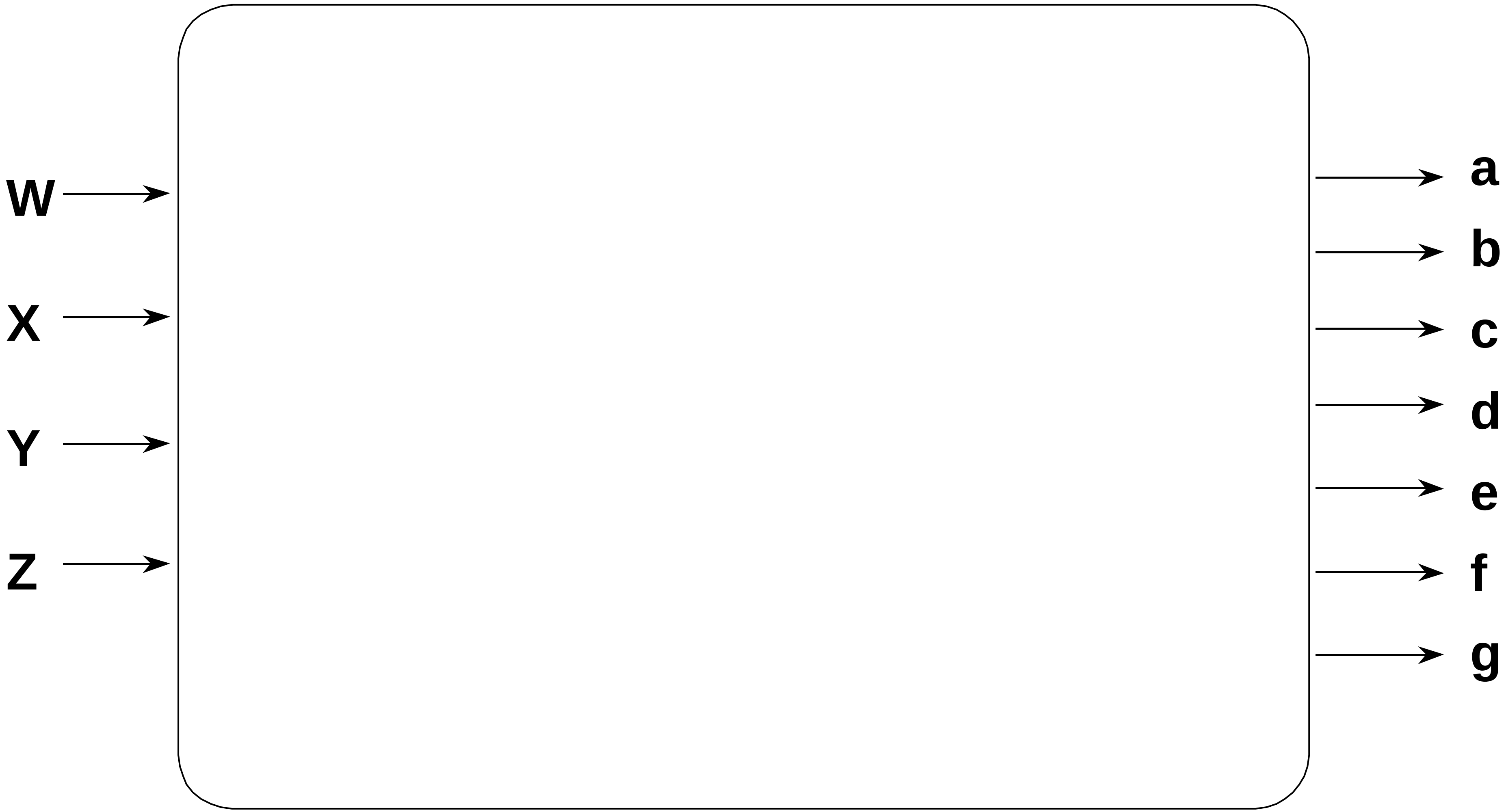






W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

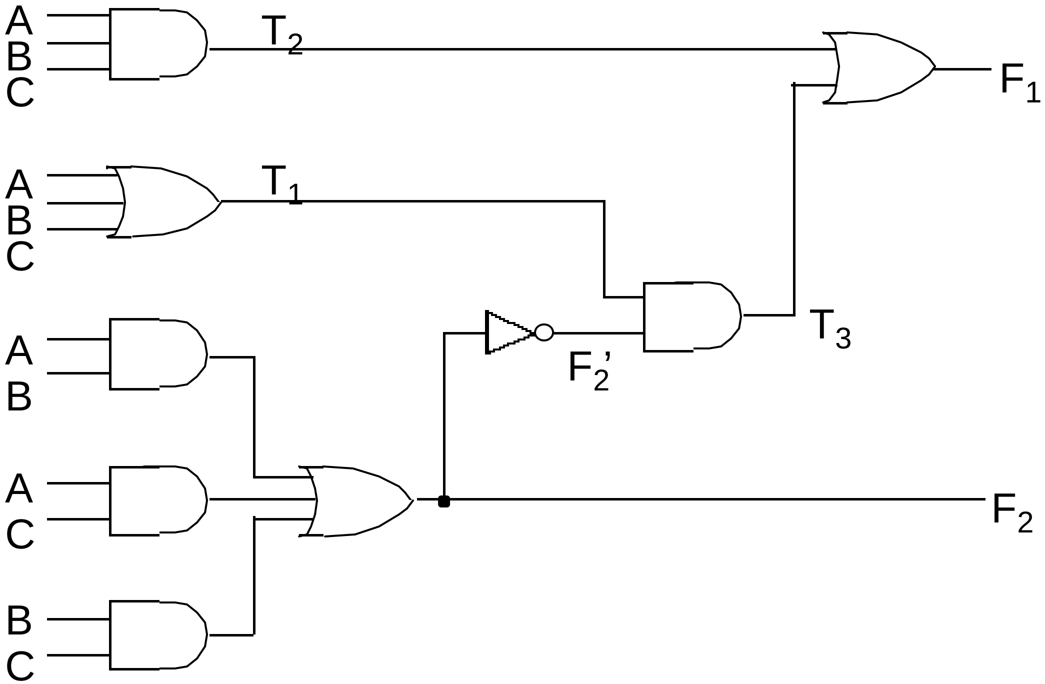
BCD to 7-Segment Display의 회로도



Analysis of Combinational Circuit

(1)부울대수식유도

- 입력변수로 부터 각 게이트의 출력에 중간변수 할당
- 중간변수에 대한 부울대수식 유도
- 회로의 출력에 대한 부울대수식을 얻을때까지 반복
- 중간변수를 대입하여 최종출력을 입력변수에 대하여 표현



$F_2 = AB+BC+CA$

$T_1 = A+B+C$

$T_2 = ABC$

$T_3 = F_2' T_1$

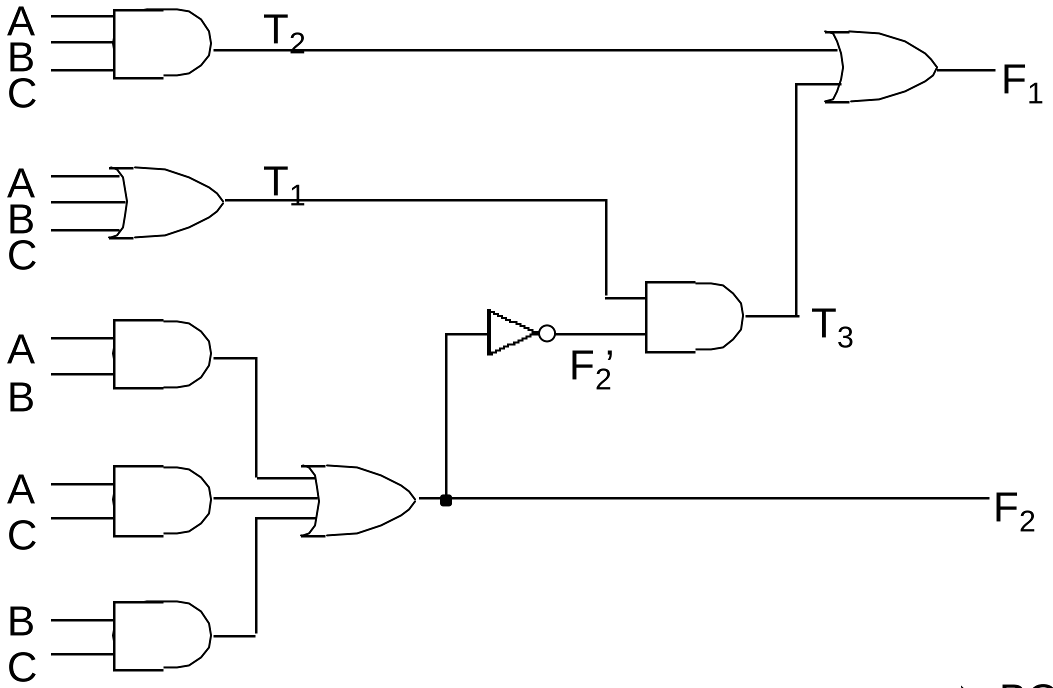
$F_1 = T_2+T_3$

$$
\begin{aligned}
F_1 &= T_2+T_3 \\
&= ABC+F_2' T_1 \\
&= ABC+(AB+BC+CA)'(A+B+C) \\
&= A'BC'+A'B'C+AB'C'+ABC \\
&= A \oplus B \oplus C
\end{aligned}
$$

jhlee@hnu.ac.kr

Analysis of Combinational Circuit

(2) **진리표작성후** map**을 이용**



A	B	C	F_2	F_2'	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	0
0	1	0	0	1	1	0	1	0
0	1	1	1	0	1	0	0	1
1	0	0	0	1	1	0	1	0
1	0	1	1	0	1	0	0	1
1	1	0	1	0	1	0	0	1
1	1	1	1	0	1	1	0	1

A \ BC	00	01	11	10
0		1		1
1	1		1	

$F_1 = A \oplus B \oplus C$

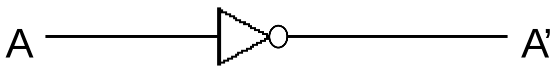
A \ BC	00	01	11	10
0			1	
1		1	1	1

$F_2 = AB + BC + CA$

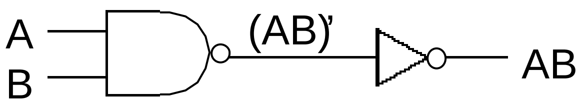
jhlee@hnu.ac.kr

Multi-Level NAND Circuit

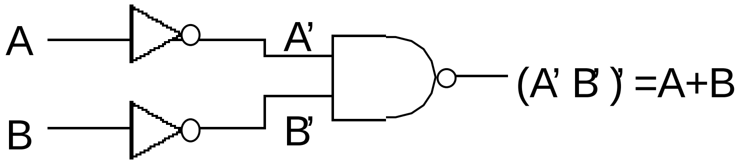
NOT

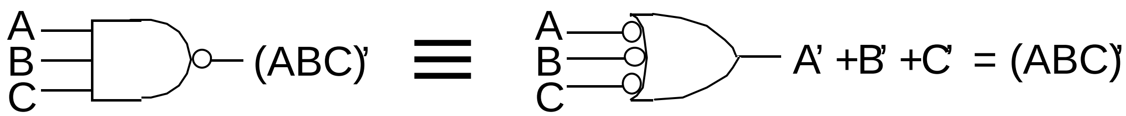


AND



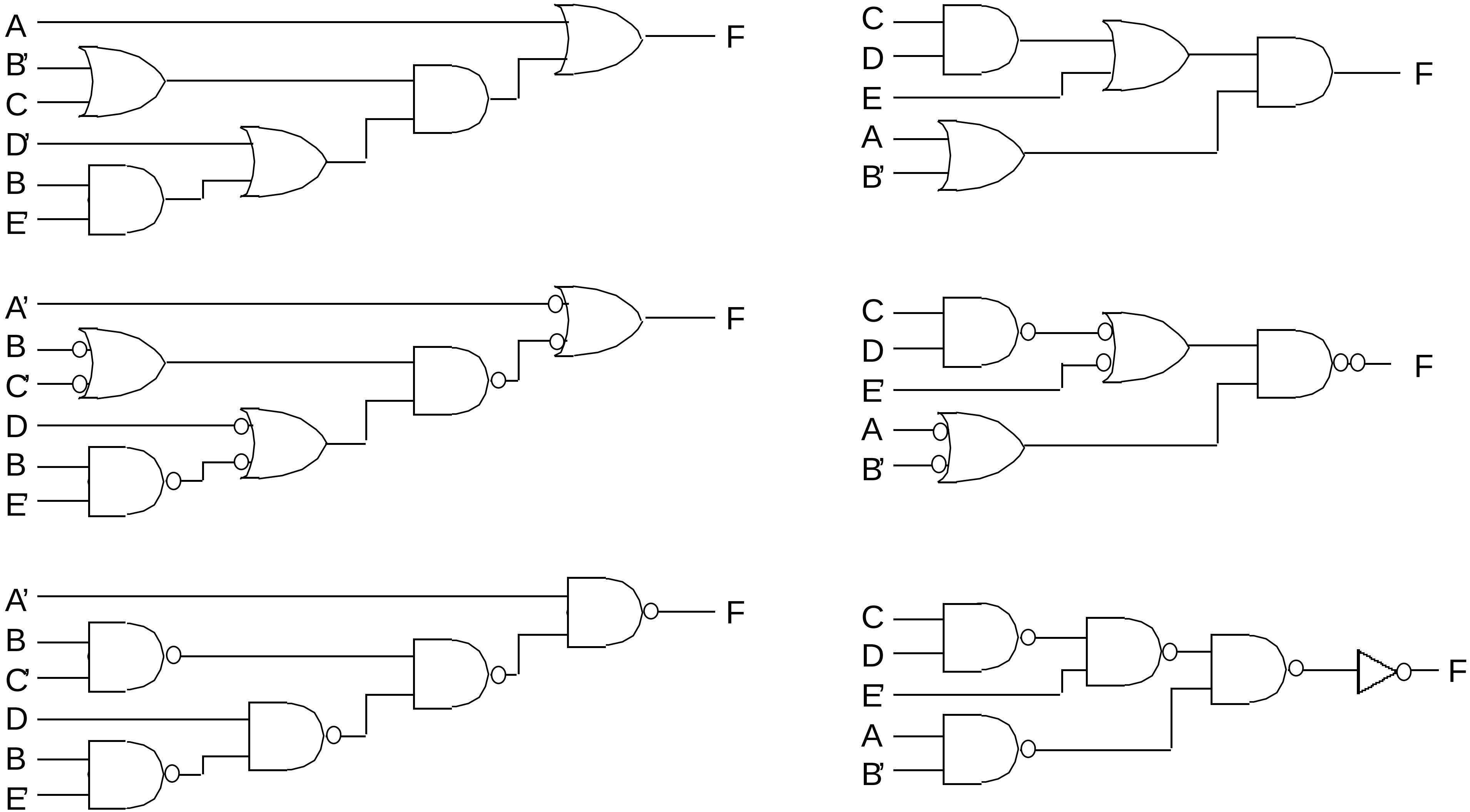
OR





jhlee@hnu.ac.kr

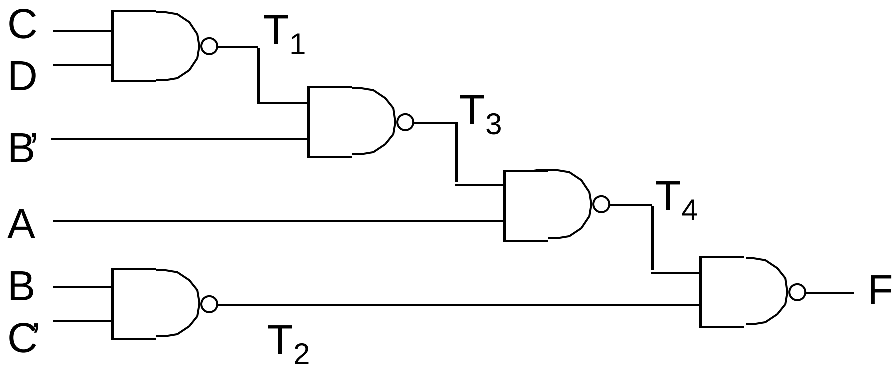
Multi-Level NAND Circuit의 구현



jhlee@hnu.ac.kr

Analysis of Multi-Level NAND Circuit

(1) 부울대수식유도



$T_1 = (CD)' = C' + D'$

$T_2 = (BC')' = B' + C$

$T_3 = (T_1 B')' = [(C' + D')B']' = (B'C' + B'D')'$

$= (B+C)(B+D) = B+CD$

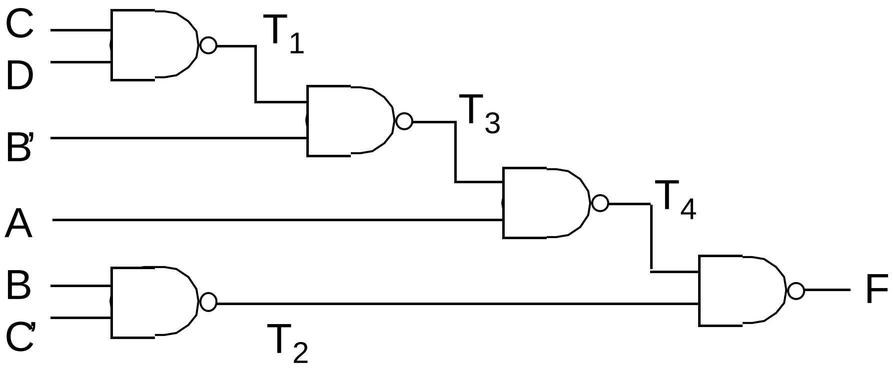
$T_4 = (AT_3)' = [(A(B+CD)]'$

$F = (T_2 T_4)' = (BC')[A(B+CD)]'$

$= BC' + A(B+CD)$

Analysis of Multi-Level NAND Circuit

(2) Truth table작성후 map을 이용



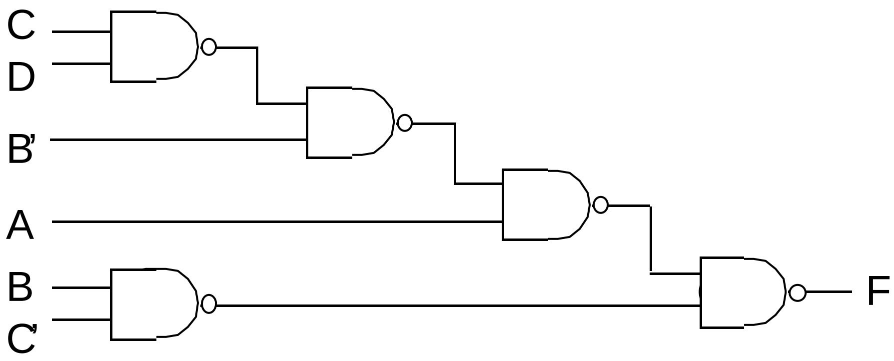
AB \ CD	00	01	11	10
00				
01	1	1		
11	1	1	1	1
10			1	

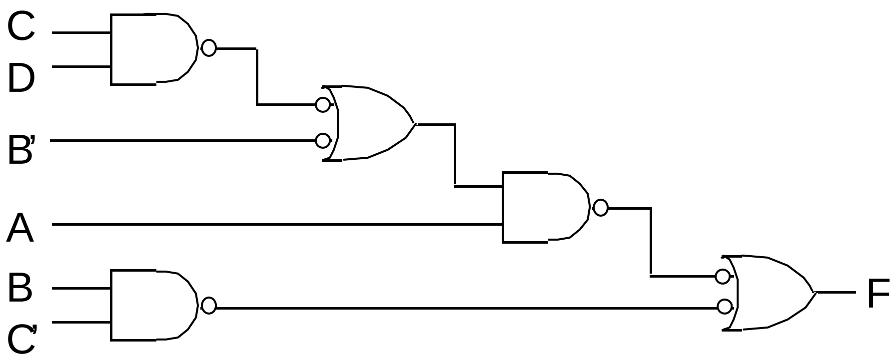
$$F = AB + BC' + ACD$$
$$= A(B + CD) + BC'$$

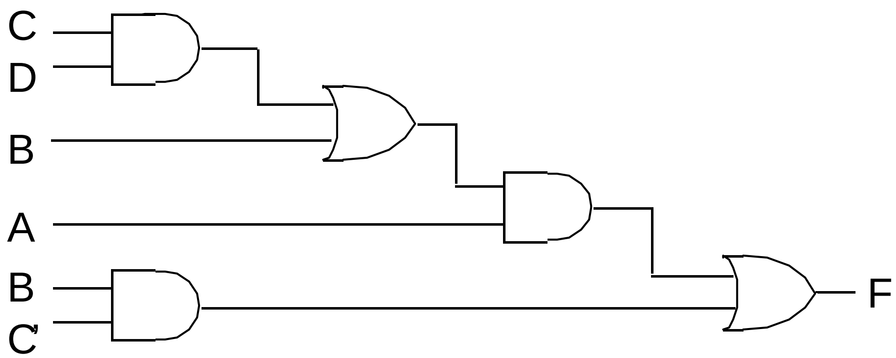
A	B	C	D	T_1	T_2	T_3	T_4	F
0	0	0	0	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	1	0	1	1	0	0	0
0	0	1	1	0	1	1	1	0
0	1	0	0	1	0	1	1	1
0	1	0	1	1	0	1	1	1
0	1	1	0	1	1	1	1	0
0	1	1	1	0	1	1	1	0
1	0	0	0	1	1	0	0	0
1	0	0	1	1	1	0	0	0
1	0	1	0	1	1	0	0	0
1	0	1	1	0	1	1	1	1
1	1	0	0	1	0	1	1	1
1	1	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1	1
1	1	1	1	0	1	1	1	1

Analysis of Multi-Level NAND Circuit

(3) AND-OR 회로도로 변환



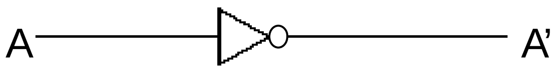




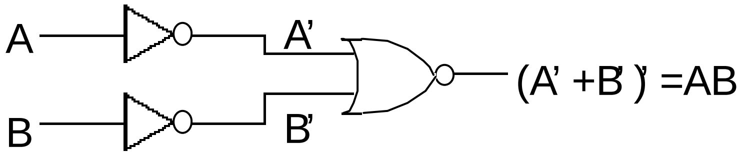
F = A(B+CD)+BC'

Multi-Level NOR Circuit

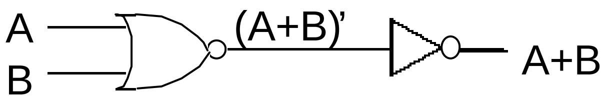
NOT

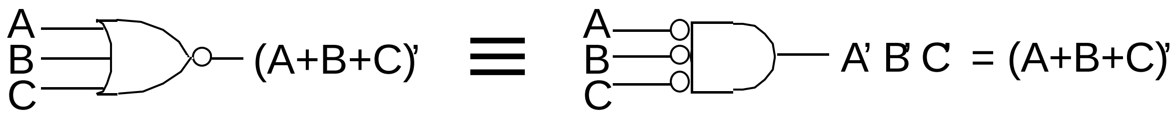


AND

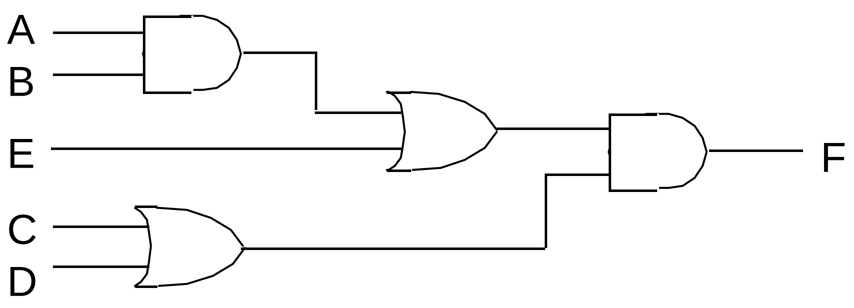


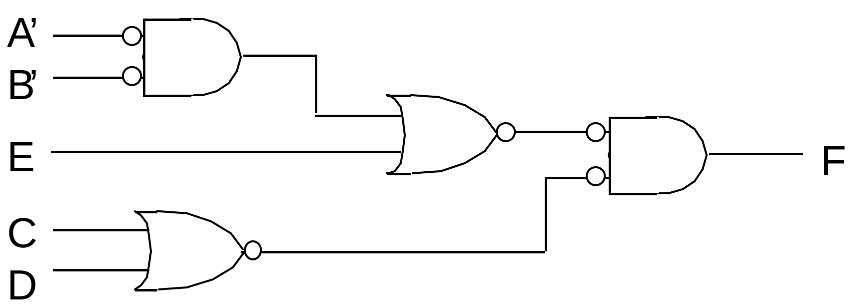
OR

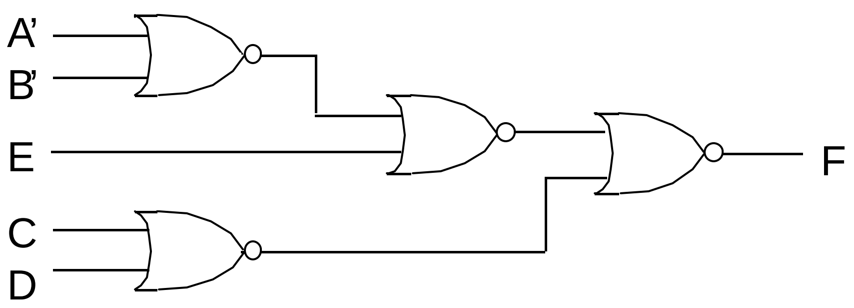




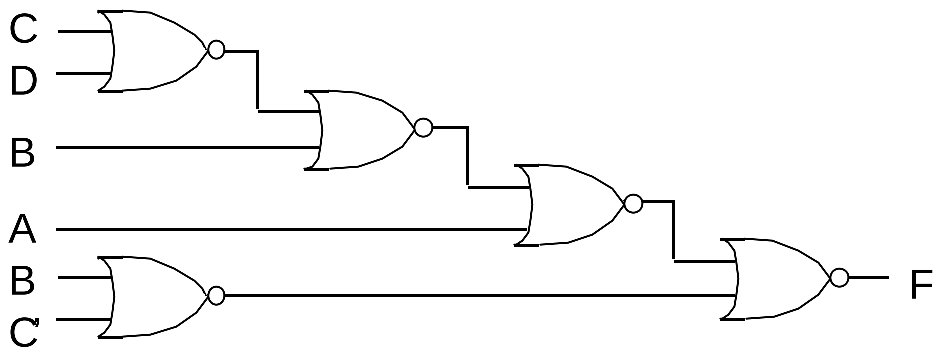
Multi-Level NOR Circuit의 구현

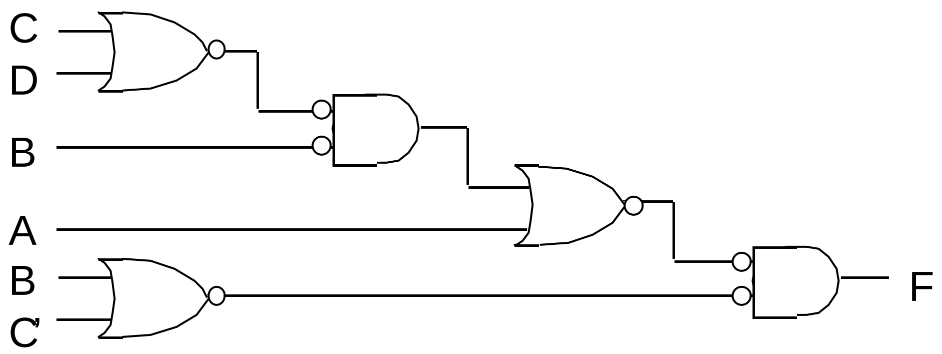


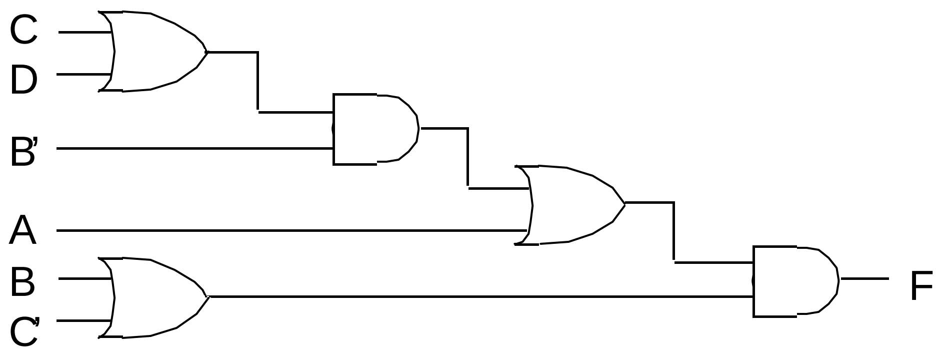




Multi-Level NOR Circuit의 해석







XOR function

$x\oplus y = x'y+xy'$

$x\oplus 0 = x$

$x\oplus 1 = x'$

$x\oplus x = 0$

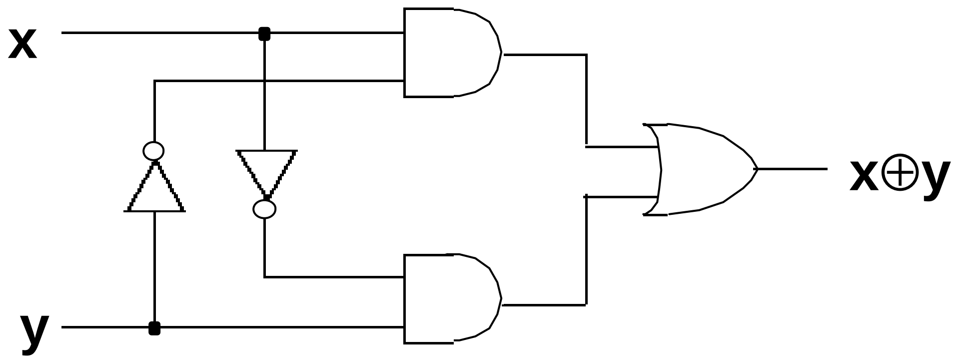
$x\oplus x' = 1$

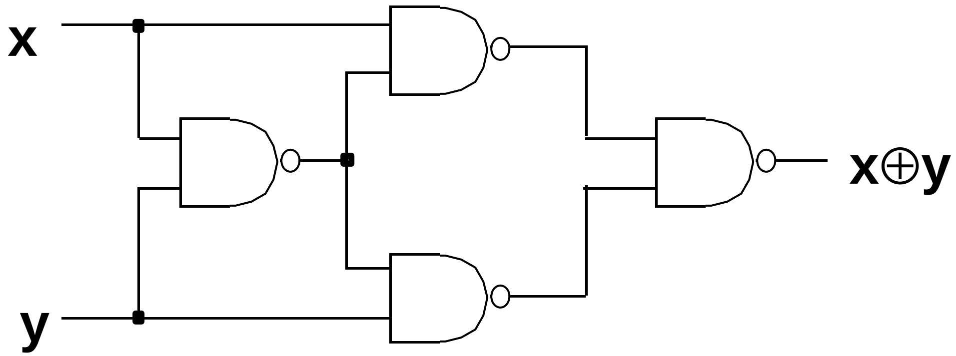
$x\oplus y' = (x\oplus y)' = x\odot y$

$x'\oplus y = (x\oplus y)' = x\odot y$

$x\odot y' = (x\odot y)' = x\oplus y$

$x'\odot y = (x\odot y)' = x\oplus y$





Odd function

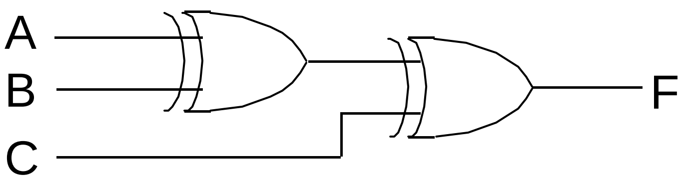
"1" 의 갯수가 홀수일때 "1" 출력

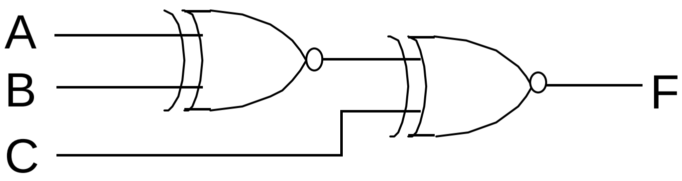
A \ BC	00	01	11	10
0		**1**		**1**
1	**1**		**1**	

$F = A \oplus B \oplus C$

$= A \odot (B \oplus C)'$

$= A \odot B \odot C$





Even function

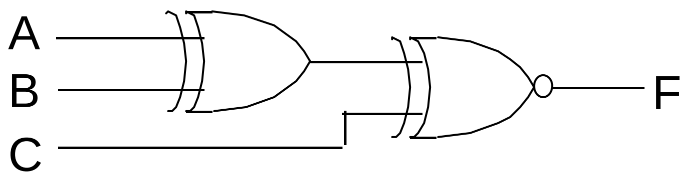
"1" 의 갯수가 짝수일때 "1" 출력

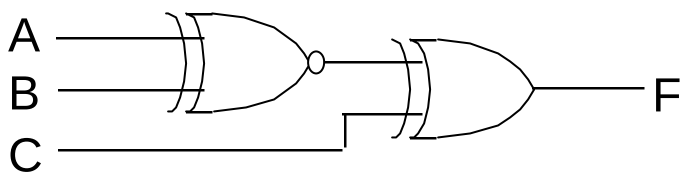
A \ BC	00	01	11	10
0	**1**		**1**	
1		**1**		**1**

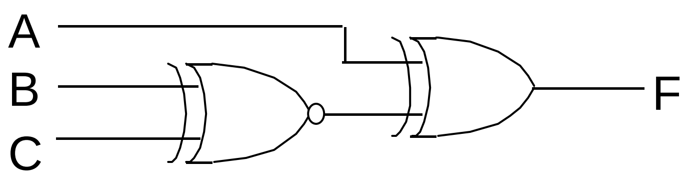
$F = (A \oplus B \oplus C)'$

$= A \oplus (B \oplus C)' = A \oplus (B \odot C)$

$= (A \oplus B)' \oplus C = (A \odot B) \oplus C$







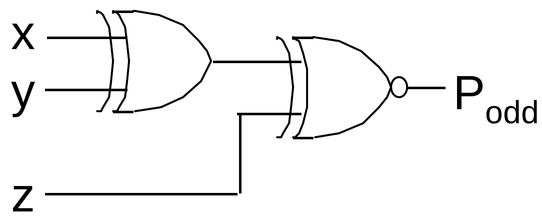
Parity Generator

x	y	z	P_{odd}	P_{even}
0	0	0	1	0
0	0	1	0	1
0	0	0	0	1
0	1	1	1	0
1	1	0	0	1
1	0	1	1	0
1	0	0	1	0
1	1	1	0	1

Odd parity

A \ BC	00	01	11	10
0	**1**		**1**	
1		**1**		**1**

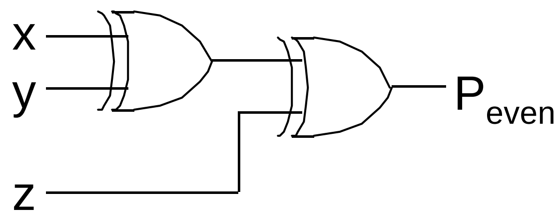
$$P_{odd} = (x \oplus y \oplus z)'$$
$$= x \oplus (y \odot z)$$
$$= (x \odot y) \oplus z$$



Even parity

A \ BC	00	01	11	10
0		**1**		**1**
1	**1**		**1**	

$$P_{even} = x \oplus y \oplus z$$
$$= [x \oplus y \oplus z]''$$
$$= [x \oplus (y \odot z)]'$$
$$= x \odot y \odot z$$



(cf) $(x \oplus)y \odot z = x \oplus (y \odot z)$

Parity Checker

x	y	z	P	C_{odd}	C_{even}
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

Odd parity

xy \ zP	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

$$C_{odd} = (x \oplus y \oplus z \oplus P)'$$
$$= x \odot (y \oplus z \oplus P)$$
$$= x \odot y \odot (z \oplus P)'$$
$$= x \odot y \odot z \odot P$$

Even parity

xy \ zP	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$$C_{even} = x \oplus y \oplus z \oplus P$$

jhlee@hnu.ac.kr